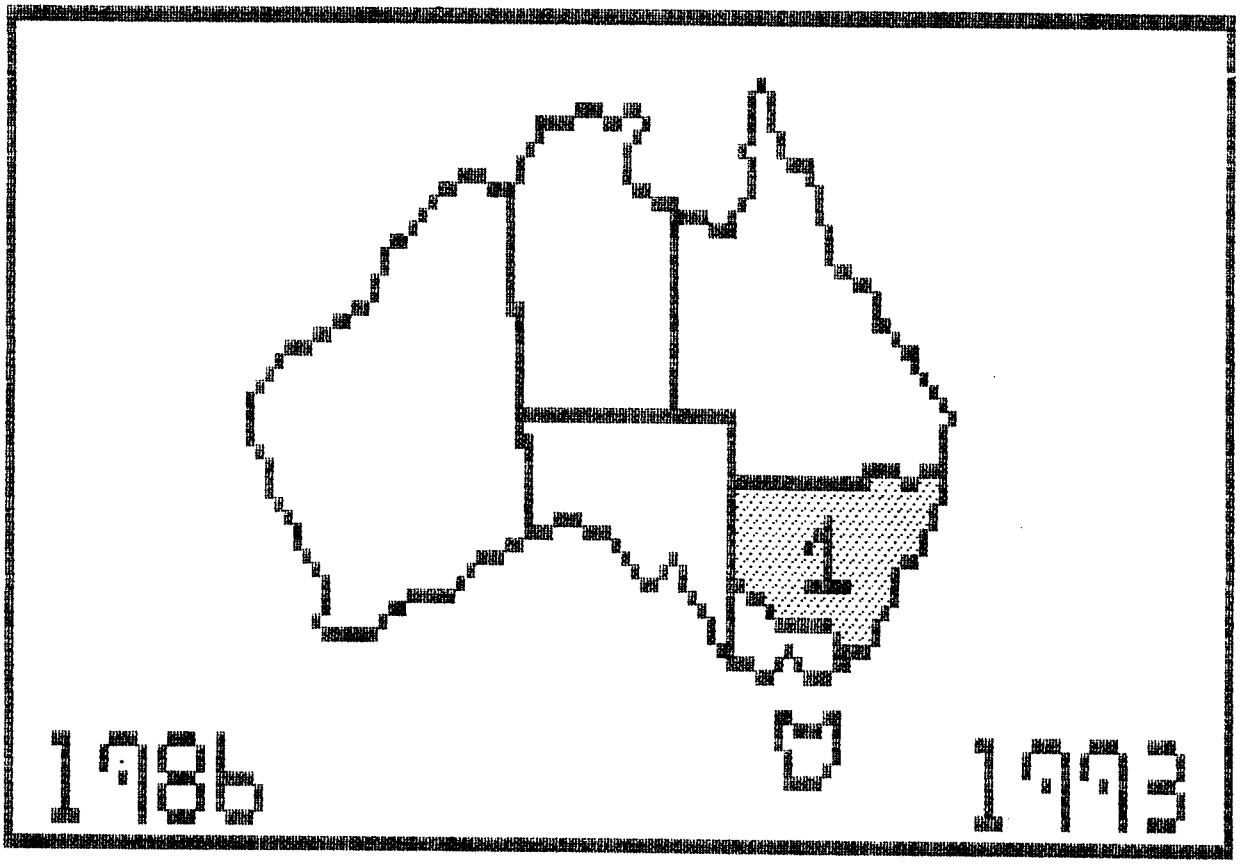
JUNE 1993 ISSUE # 43

HUNTER VALLEY VZ JOURNAL

HAPPY 7TH BIRTHDAY



PRODUCED BY JOSEPH P. LEON
A NON PROFIT PUBLICATION

FRONT COVER

MY THANKS TO JASON OAKLEY FOR DESIGNING TOP HALF WHILE THE REST IS SELF EXPLANATORY. FOR THOSE WHO CAN'T WORK OUT WHAT THE 1 STANDS FOR THEN TURN TO PAGE 3.

HELP — SELL & TELL PAGE 3

7TH BIRTHDAY, APOLOGIES, NEXT/FUTURE AND BACK ISSUES.

BUGS REPORT FOR DOS V1.2 BY LESLIE MILBURN PAGE 4

LESLIE HAS FOUND SOME BUGS IN DOS AND IS SHARING IT WITH OTHER DOS USERS. THANKS LESLIE.

DISABLING BREAK KEY BY LESLIE MILBURN PAGES 4—8

LESLIE SHOWS US HOW TO CREATE BREAK KEY PROOF BASIC PROGRAMS. IF YOU HAVE WANTED TO LOCK UP YOUR PROGRAMS FOR WHATEVER REASON YOU CAN NOW DO SO.

INTRODUCTION TO PROGRAMMING PART II BY BOB KITCH PAGES 8—11

BOB CONCLUDES HIS ESSAY INTO MORE STRUCTURED PROGRAMMING AND MOST OF US SHOULD BENEFIT FROM HIS EFFORTS.

WP FILE RETRIEVE BY DAVE MITCHELL PAGES 11—15

THE ABOVE DISK UTILITY HAS BEEN DESIGNED TO RESURRECT WP FILE DISKS WHOSE TRACK ZERO ARE UNREADABLE. I FOUND OUT IT CAN DO MORE THAN THAT.

PREVENTITIVE DISK MAINTAINANCE PAGE 15

DAVE MITCHELL SOFTWARE FOR SALE PAGE 16

E&F DISK WP PATCH3.3
DISK FILER CATALOGUER
EXTENDED DOS VERSION 1.3
MENU/FILE COPIER

USER GROUPS — CONTRIBUTIONS SUBSCRIPTIONS PAGE 16

HUNTER VALLEY VZ JOURNAL INDEX PART I — SUPPLEMENT PAGES 01—05

THE LAST INDEX COVERING ISSUES 1 TO 24 WAS PUBLISHED IN ISSUE 24, MAY/JUNE 1989. THIS CURRENT INDEX COVERS ISSUES 1 TO 20 WITH 21 TO 44 PLANNED FOR NEXT ISSUE. THE REASON FOR REPEATING THE FORMER IS THAT THE MAJORITY OF REMAINING SUBSCRIBERS WERE NOT MEMBERS AT THAT TIME.

DISCLAIMER: EVERY EFFORT IS MADE TO INSURE THE ACCURACY OF INFORMATION CONTAINED WITHIN BE IT GENERAL, TECHNICAL, PROGRAMMING, ETC. NO RESPONSIBILITY CAN BE ACCEPTED BY HUNTER VALLEY VZ USERS' GROUP OR AUTHOR AS A RESULT OF APPLYING SUCH INFORMATION IN PRACTICE.

COPYRIGHT: THE HUNTER VALLEY VZ JOURNAL IS SUBJECT TO COPYRIGHT AND NO MATERIAL IN THE JOURNAL MAY BE REPRODUCED IN PART OR WHOLE WITHOUT THE CONSENT OF THE HUNTER VALLEY USERS' GROUP OR THE AUTHOR WHO RETAINS COPYRIGHT.

DOS BUGS REPORT FOR DOS V1.2 43-4

BY LESLIE MILBURN

1. IF YOU USE CLOSE IN IMMEDIATE MODE, THE FILE BUFFER IS NOT FLUSHED TO DISK.

2. WHEN SAVING STRINGS TO DISK USING PR#, THE STRING MUST NOT CONTAIN A COMMA, COLON OR CARRIAGE RETURN.

3. THE 116 BYTES OF "UNUSED MEMORY" IN THE COMMUNICATIONS REGION (7A29H - 7A9CH) IS USED BY THE IN# COMMAND. THIS MEANS THAT KSCAN, MOUSE DRIVER 1.04 AND FIND WILL BE DESTROYED IF A BASIC PROGRAM USING IN# IS RUN.

4. THE MAXIMUM NUMBER OF CHARACTERS (BYTES) THAT CAN BE READ BY ONE IN# COMMAND IS 199. THIS IS REGARDLESS OF THE NUMBER OF VARIABLE ARGUEMENTS.

5. DO NOT USE ANY OTHER DISK COMMANDS WHILE A DATA FILE IS OPEN, OTHER THAN OPEN, CLOSE, IN# AND PR#, OTHERWISE CORRUPT DATA COULD RESULT.

6. WHEN A BASIC PROGRAM PERFORMS A LOT OF STRING MANIPULATION, THE "GARBAGE COLLECTOR" OCCASIONALLY TAKES CONTROL AND REARRANGES THE BASIC STRING SPACE. IF THIS HAPPENS WHILE YOU ARE SAVING STRINGS TO DISK VIA PR#, A CORRUPT DATA FILE WILL OCCUR AS THE STRINGS HAVE BEEN MOVED WITHOUT PR# KNOWING ABOUT IT.

IF ANYBODY HAS FOUND ANY OTHER BUGS IN ANY DOS SYSTEM, PLEASE INFORM THE EDITOR WHO CAN PASS THE INFORMATION ON TO OTHER USERS.

CREATING BREAK KEY PROOF BASIC PROGRAMS BY LESLIE MILBURN

WHEN I FIRST BOUGHT MY VZ NEARLY 9 YEARS AGO, ONE OF THE FIRST THINGS I WANTED TO DO WAS TO DISABLE THE BREAK KEY TO STOP PEOPLE LISTING MY BASIC PROGRAMS. WHEN I ENQUIRED FURTHER I WAS TOLD THAT IT COULD NOT BE DONE. HOWEVER, IT CAN AND FAIRLY EASILY!

ABOUT THE BREAK KEY

THE MAIN PROBLEM WHEN DEALING WITH THE BREAK KEY IS THAT MANY OF THE ROM ROUTINES CHECK FOR IT IN DIFFERENT WAYS.

THE ORIGINAL LEVEL II BASIC ROM ROUTINES CHECK THE BREAK KEY VIA TWO MAIN ROUTINES AND REPLACING THESE IS NOT DIFFICULT. KSCAN (REFER JOURNAL #38) DOES EXACTLY THAT.

UNFORTUNATELY, THE ROM ROUTINES WHICH HAVE BEEN ADDED SPECIFICALLY FOR THE VZ (I.E. EXTENSIONS TO THE LEVEL II BASIC) HAVE THE BREAK KEY CHECKS IN ROM AND CANNOT BE INTERCEPTED. THESE ROUTINES CORRESPOND TO THE FOLLOWING BASIC COMMANDS:-

CLOAD, COPY, CRUN, CSAVE, INPUT#, PRINT#, LPRINT, SOUND, VERIFY

NOTICE THAT ALL OF THESE COMMANDS DEAL WITH EITHER CASSETTE, PRINTER OR SPEAKER I/O. ALSO NOTE THAT IF YOU HAVE A DISK DRIVE ATTACHED, ALL DISK BASIC COMMANDS CHECK THAT THE MINUS KEY IS PRESSED (NOT CTRL-MINUS). THIS MAY NOT BE THE CASE FOR YOUR DISK CONTROLLER AS VARIATIONS EXIST.

AS YOU CAN SEE, THIS IS VERY SIMILAR TO THE PREVIOUS M/C ROUTINE EXCEPT THAT THE BREAK KEY ROUTINE IS INSTALLED PRIOR TO RUNNING THE BASIC PROGRAM. THIS MEANS THAT THE BREAK KEY ROUTINE MUST BE SAVED ALONG WITH THE HEADER & BASIC PROGRAM. SEE BELOW FOR FURTHER DETAILS.

USING THIS TECHNIQUE WE CAN NOW SAVE A BASIC PROGRAM AS AN AUTO-EXECUTING PROGRAM WHICH DISABLES THE BREAK KEY. IMMEDIATE MODE CAN ONLY BE ENTERED IN THREE CASES:-

(1) THE BREAK KEY WAS PRESSED WHILE THE PROGRAM WAS PERFORMING CASSETTE, PRINTER, SPEAKER OR DISK I/O

(2) A COMMAND WAS EXECUTED WHICH RETURNS TO IMMEDIATE MODE. THESE COMMANDS ARE:-

LIST, LLIST, STOP, END & NEW

(3) AN ERROR OCCURRED AND NO ERROR HANDLER WAS INSTALLED.

(I.E. NO ON ERROR COMMAND WAS INCLUDED IN THE PROGRAM).

THEREFORE, YOUR PROGRAM CANNOT BE LISTED ONCE RUN UNLESS ONE OF THE ABOVE CONDITIONS OCCUR.

WHAT IF THE PROGRAM IS BLOAD'ED NOT BRUN?

AS MENTIONED ABOVE, THE BREAK KEY ROUTINE MUST BE SAVED ALONG WITH THE BASIC PROGRAM AND M/C HEADER. THE TECHNICAL REFERENCE MANUAL DISCUSSES THREE METHODS FOR RESERVING MEMORY FOR A MACHINE CODE SUBROUTINE. TWO OF THESE ARE SUITABLE FOR OUR PURPOSE.

ONE METHOD INVOLVES INCREASING THE END OF BASIC POINTER AND SAVING THE M/C ROUTINE IN THE GAP CREATED AFTER THE BASIC PROGRAM. ONE DISADVANTAGE OF THIS METHOD IS THAT AS THE BASIC PROGRAM GROWS, THE M/C START ADDRESS CHANGES.

THE OTHER METHOD INVOLVES MOVING THE START OF BASIC POINTER THUS RESERVING MEMORY BETWEEN THE END OF THE COMMUNICATIONS REGION AND THE NEW START OF BASIC.

THE LATTER METHOD IS MORE FAVOURABLE FOR AN IMPORTANT REASON, WHAT IF THE BREAK PROTECTED PROGRAM IS BLOAD'ED NOT BRUN. IF THIS OCCURS THE PROGRAM CAN BE LISTED BUT NOT ALTERED AS THE START AND END OF BASIC POINTERS ARE NOT CORRECT. TO PREVENT THE PROGRAM FROM BEING ABLE TO BE LISTED WE MUST FOOL THE BASIC INTERPRETER INTO BELIEVING THAT THE PROGRAM AREA IS EMPTY.

THIS IS DONE BY SETTING THE FIRST TWO BYTES IN THE PROGRAM AREA TO NULL BYTES. AS MOST USERS DO NOT BOTHER TO ALTER THE START OF BASIC POINTER FROM ITS DEFAULT VALUE WE CAN ASSUME THAT IT IS SET TO 31465 AND STORE NULL IN 31465 & 31466.

WE CAN NOW LIST THE STEPS TO CREATE A "BREAK KEY PROOF, AUTO-STARTING, NON-LISTABLE" BASIC PROGRAM. THESE ARE AS FOLLOWS:-

(1) INCREASE THE START OF BASIC POINTER AT 30884/5 SUFFICIENTLY TO STORE THE M/C ROUTINES.

(2) TYPE IN THE BASIC PROGRAM. NOTE THAT THE PROGRAM CANNOT BE LOADED FROM TAPE OR DISK WITHOUT THE AID OF A UTILITY. THIS IS BECAUSE PROGRAMS ARE ALWAYS LOADED BACK TO THE SAME ADDRESS THAT THEY WERE SAVED FROM.

```
071 ;INTR: NEW INTERRUPT
072 ;      ROUTINE.
073 ;
074 ;      NOTE: REGISTERS HAVE
075 ;            BEEN SAVED UPON
076 ;            ENTRY TO THIS
077 ;            FUNCTION.
078 ;
079 INTR CALL 3F7BH
080 ;DISPLAY THE CURSOR
081      CALL 2EDCH
082 ;SCAN THE KEYBOARD
083      CALL 002BH
084 ;SAVE THE KEY CODE
085      PUSH AF
086 ;IF IN INPUT PHASE THEN
087 ;DISPLAY CHARACTER, UPDATE
088 ;SCREEN POSITION, ETC.
089      LD   HL,7839H
090      BIT  0,(HL)
091      CALL Z,301BH
092 ;RESTORE KEY CODE.
093      POP  AF
094 ;SOUND A BEEP IF KEY WAS
095 ;PRESSED.
096      CALL 3430H
097 ;REMOVE RETURN ADDRESS SO
098 ;THAT THE ROM ROUTINE IS
099 ;BY-PASSED.
100      POP  HL
101 ;RESTORE PREVIOUSLY SAVED
102 ;REGISTERS.
103      POP  HL
104      POP  DE
105      POP  BC
106      POP  AF
107 ;ENABLE INT'S AND RETURN.
108      EI
109      RETI
110 ;**************************
111 ;SCAN: THIS FUNCTION SCANS
112 ;      THE KEYBOARD.
113 ;
114 SCAN CALL 2EFDH
115      CP   1
116      RET  NZ
117      LD   A,0
118      RET
119 END  EQU  $
```

INTRODUCTION TO PROGRAMMING PART II BY BOB KITCH

AS MENTIONED IN PART 1 OF THIS SERIES, THE PROGRAMMING TASK IS A LARGE AND COMPLEX FEAT OF ORGANIZATION AND REQUIRES A WIDE RANGE OF SKILLS. IT IS POSSIBLE, AND BEST, TO BREAK THE TASK DOWN INTO SIX SEGMENTS - EACH OF WHICH MUST BE THOUGHT ABOUT, PLANNED AND THEN CARRIED OUT TO ENSURE THE SUCCESSFUL COMPLETION OF A SOFTWARE PROJECT.

EVEN A SMALL PROGRAM REQUIRES THAT A CURSORY CONSIDERATION OF THE SIX SEGMENTS BE MADE - ALTHOUGH SOME OF THEM MAY BE QUICKLY PASSED OVER AS TRIVIAL. BUT IT IS CERTAIN THAT LARGER PROGRAMS (MORE THAN 200 LINES) REQUIRE CAREFUL PLANNING FOR SUCCESS.

BEFORE DESCRIBING THE SIX STEPS, IT IS WORTH THINKING ABOUT "WHAT MAKES A GOOD PROGRAM?"

A PROGRAM MAY BE JUDGED FROM A NUMBER OF DIFFERENT STANDPOINTS; EACH IS NOT NECESSARILY MUTUALLY EXCLUSIVE AND SOMETIMES SOME CONFLICTS REQUIRE THAT A TRADE-OFF BE MADE.

THE FIRST CRITERIA IS THAT A PROGRAM SHOULD BE EFFICIENT. EFFICIENCY CAN BE CONSIDERED FROM A NUMBER OF VARYING VIEW POINTS. FOR EXAMPLE, OPTIMIZATION OF THE RUN-TIME CAN BE CONSIDERED AS EFFICIENT. ALSO, REDUCTION IN STORAGE REQUIREMENTS FOR BOTH PROGRAM CODE AND VARIABLES CAN BE CONSIDERED AS EFFICIENT PROGRAMMING.

FURTHERMORE, AND PARTICULARLY IF ONE IS DEVELOPING SOFTWARE COMMERCIALLY, THEN EFFICIENCY CAN BE MEASURED IN TERMS OF THE ACTUAL TIME REQUIRED TO GET AN APPLICATIONS PROGRAM RUNNING AND THE EASE OF MAINTENANCE OF THAT CODE. THE USE OF APPROPRIATE DATA TYPES AND DATA STRUCTURES CAN GREATLY IMPROVE THE EFFICIENCY OF A PROGRAM. THE SELECTION OF A SUITABLE ALGORITHM CAN ALSO ASSIST. FINALLY, EASE OF DEBUGGING SO THAT THE PROGRAM CAN BE UPDATED OR MODIFIED MAY BE CONSIDERED DESIRABLE.

NEXT, DEFINE THE OUTPUT EXPECTED FROM THE PROGRAM. IS IT TO WRITE TO TAPE AND IN WHAT FORMAT? PERHAPS IT IS TO BE SCREEN ORIENTED - CAN SOUND BE USED - OR PERHAPS VOICE SYNTHESIS TO TELL THE OPERATOR WHAT IS GOING ON? PLAN VERY CAREFULLY AND FULLY THE LAYOUT OF THE EXPECTED OUTPUT AS THIS IS HOW USERS WILL INITIALLY PERCEIVE THE QUALITY OF THE PROGRAM.

AFTER DEFINING THE I/O FOR THE PROGRAM WE SHOULD NOW HAVE A FEEL FOR THE ANTICIPATED RANGE OF PARAMETERS THAT THE PROGRAM IS MEANT TO ACCEPT AND ALSO HANDLE. THIS BRINGS IN THE VERY IMPORTANT CONCEPT OF DEFINING THE BOUNDS WITHIN WHICH THE PROGRAM MUST FUNCTION CORRECTLY. FOLLOWING ON FROM THIS, IS RANGE CHECKING OF ALL INPUT PARAMETERS SO THAT THE PROGRAM CANNOT GO BEYOND THE RANGE THAT IT WAS DESIGNED FOR AND GIVE UNEXPECTED RESULTS.

A NUMBER OF WARNING MESSAGES MUST BE BUILT INTO THE PROGRAM ALONG WITH ERROR CAPTURE AND RECOVERY ROUTINES. IT IS FAILURE TO DEFINE THE OPERATING BOUNDS OF A PROGRAM THAT CAUSES MOST CRASHES OR ROGUE BEHAVIOUR. EVEN THE DEFINITION OF INTEGER VARIABLES AT THIS STAGE CAN ASSIST BY IMPROVING PROGRAM EXECUTION TIME AND REDUCING STORAGE REQUIREMENTS.

THE DEFINITION STAGE SHOULD BE ROUGHED OUT ON PIECES OF PAPER KEPT FOR LATER REFERENCE. PERHAPS BETTER, IS TO USE AN OLD EXERCISE BOOK. ANOTHER BENEFIT OF THIS IS THAT OVER A PERIOD OF MONTHS YOUR PROGRESS CAN BE MEASURED AND YOUR GROWTH OF PROGRAMMING IDEAS RECORDED. ANOTHER BENEFIT (ALTHOUGH I HARDLY DARE MENTION IT!) IS THAT IF, AFTER THE CODING STAGE, A SYSTEM CRASH OCCURS AND YOU DIDN'T SAVE THE PROGRAM, THEN ALL IS NOT LOST - AT LEAST AN OUTLINE OF THE PROGRAM REMAINS.

4. DESIGN PHASE

HAVING SORTED OUT I/O AND OPERATING BOUNDS, THE ACTUAL SELECTION OF AN ALGORITHM TO ACHIEVE THE RESULT IS COMMENCED. BY THIS TIME SOME IDEA OF THE NUMBER OF VARIABLES REQUIRED AND THEIR TYPE SHOULD HAVE BEGUN TO GEL.

THIS IS ALSO THE STAGE WHERE YOUR BASIC HONESTY IN STAGES 1 AND 2 MAY CATCH UP WITH YOU! DATA STRUCTURE ORGANISATION AND ALGORITHM SELECTION ARE REALLY EXPERIENCE-RELATED SKILLS - HENCE THE SUGGESTION TO READ AND/OR MODIFY EXISTING PROGRAMS. BUT DO NOT DESPAIR - PRACTICE MAKES PERFECT.

5. IMPLEMENTATION PHASE

TO DATE VERY LITTLE ACTUAL CODING SHOULD HAVE BEEN DONE; IN FACT THE COMPUTER NEED NOT EVEN HAVE BEEN TURNED ON! SOME PEOPLE MAY BE SURPRISED AT HOW LATE IN THE TASK THE COMPUTER ACTUALLY ENTERS INTO THE PICTURE. AN AWFUL LOT OF PLANNING AND ORGANIZING CAN BE DONE OFF THE COMPUTER AND ON THE "BACKS OF OLD ENVELOPES".

IT IS ALSO AT THIS STAGE THAT THE CHOICE OF PROGRAMMING LANGUAGE SHOULD BE MADE. IS THE PROGRAM TIME DEPENDANT? IF IT IS, THEN IT SHOULD PROBABLY BE WRITTEN IS ASSEMBLER.

IF THE ACTUAL TIMING IS NOT SO CRITICAL THEN WRITING IN BASIC WITH ITS DIAGNOSTICS AND HELPFUL FEATURES (SO TYPICAL OF A HIGH LEVEL LANGUAGE) DEEM IT SENSIBLE. EXPERIENCED PROGRAMMERS WILL PROBABLY USE A BIT OF EACH IN PRACTICE. A VERY SENSIBLE COMPROMISE IS TO DEVELOPE THE PROGRAM IN INTERPRETED BASIC AND ONCE FINALIZED AND DEBUGGED, COMPILE THE BASIC CODE TO SPEED UP EXECUTION.

```
001 ;ROUTINE FOR GETTING
002 ;E&F  WORDPROCESSOR
003 ;FILES FROM DISK WHEN
004 ;TRACK ZERO IS UNREADABLE
005       LD    HL,A0
006       LD    (788EH),HL
007 A0    LD    HL,END
008       LD    (ADD),HL
009       LD    HL,M1
010       CALL  IPUT
011       CALL  ATB
012       LD    (ADD1),A
013       LD    HL,M2
014       CALL  IPUT
015       CALL  ATB
016       LD    (ADD2),A
017       LD    HL,M4
018       CALL  IPUT
019       LD    A,20H
020       LD    (DE),A
021       PUSH  IY
022       POP   DE
023       INC   DE
024       LD    BC,8
025       LDIR
026       DI
027       CALL  4008H
028       LD    BC,0032H
029       CALL  4038H
030       CALL  4011H
031       OR    A
032       JP    NZ,400EH
033       LD    A,(ADD1)
034       LD    (IY+12H),A
035       LD    A,(ADD2)
036       LD    (IY+11H),A
037 A1    LD    L,(IY+34H)
038       LD    H,(IY+35H)
039       LD    A,(IY+12H)
040       DEC   A
041       SLA   A
042       LD    E,A
043       LD    D,0
044       LD    A,(IY+11H)
045       CP    8
046       CCF
047       ADC   HL,DE
048       AND   7
049       INC   A
050       LD    B,A
051       LD    C,(HL)
052       RLC   C
053 A13   RRC   C
054       DJNZ  A13
055       SET   0,C
056       LD    B,A
057       RRC   C
058 A14   RLC   C
059       DJNZ  A14
060       LD    (HL),C
061       CALL  4035H
062       OR    A
063       JP    NZ,400EH
064       LD    L,(IY+31H)
065       LD    H,(IY+32H)
066       LD    DE,(ADD)
067       LD    BC,007EH
068       LDIR
069       LD    (ADD),DE
070       LD    A,(HL)
071       OR    A
072       JR    Z,A2
073       LD    (IY+12H),A
074       INC   HL
075       LD    A,(HL)
076       LD    (IY+11H),A
077       JR    A1
078 A2    DEC   DE
079       LD    A,(DE)
080       OR    A
081       JR    Z,A2
082       INC   DE
083       LD    HL,END
084       EX    DE,HL
085       XOR   A
086       SBC   HL,DE
087       EX    DE,HL
088       LD    HL,0D000H
089       XOR   A
090       SBC   HL,DE
091       LD    (ADD),HL
092       CALL  402CH
093       OR    A
094       JP    NZ,400EH
095       CALL  402FH
096       OR    A
097       JP    NZ,400EH
098       LD    (HL),57H
099       INC   HL
100       LD    (HL),3AH
101       INC   HL
102       PUSH  HL
103       POP   DE
104       PUSH  IY
105       POP   HL
106       INC   HL
107       LD    BC,8
108       LDIR
109       LD    A,(ADD1)
110       LD    (DE),A
111       INC   DE
112       LD    A,(ADD2)
113       LD    (DE),A
114       INC   DE
115       LD    HL,(ADD)
116       LD    A,L
117       LD    (DE),A
118       INC   DE
119       LD    A,H
120       LD    (DE),A
121       INC   DE
122       LD    A,0
123       LD    (DE),A
124       INC   DE
```

ALL THE UTILITIES MENTIONED WILL PROVIDE THE INFORMATION REQUIRED SIMILIAR TO WHAT IS SHOWN BELOW. IF YOU'RE LIKE ME THEN YOU'LL HAVE A MIXTURE OF FILES ON THE ONE DISK. I TRIED WPRET PROCEDURE AND FOUND THAT WPRET IS MORE VERSATILE THAN I FIRST THOUGHT.

BEFORE WPRET PROCEDURE

```
T:PLOTTO-2 01 00 7AE9 98D7 1DEE
B:PARK2    04 0D 9000 90E3 00E3
B:DIRA     04 0F 7200 72E5 00E5
D:LOTTO    05 01 0000 0000 0000
B:MAPAUST  05 04 7000 7800 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

AFTER WPRET PROCEDURE

```
W:PLOTTO-2 01 00 B212 D000 1DEE
W:PARK2    04 0D CF1D D000 00E3
W:DIRA     04 0F CF1B D000 00E5
W:LOTTO    05 01 CF00 D000 0100
W:MAPAUST  05 04 C800 D000 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

AFTER FILETYPE CHANGE

```
T:PLOTTO-2 01 00 B212 D000 1DEE
B:PARK2    04 0D CF1B D000 00E3
B:DIRA     04 0F CF1B D000 00E5
D:LOTTO    05 01 CF00 D000 0100
B:MAPAUST  05 04 C800 D000 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

AFTER START ADDRESS CHANGE

```
T:PLOTTO-2 01 00 7AE9 98D7 1DEE
B:PARK2    04 0D 9000 90E3 00E3
B:DIRA     04 0F 7200 72E5 00E5
D:LOTTO    05 01 0000 0100 0100
B:MAPAUST  05 04 7000 7800 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

BEFORE WPRET PROCEDURE

THIS SHOWS DIRECTORY PRINTOUT OF DISK BEFORE EXPERIMENTATION WITH ALL INFORMATION NEEDED TO RESURECT ALL FILES.

AFTER WPRET PROCEDURE

I FORMATTTED TRACK ZERO, RAN CHKDSK AND THEN WPRET AND PRINTOUT SHOWS RESULT. PLEASE NOTE THAT ALL FILES HAVE 'W' FILETYPE AND END ADDRESS OF D000 WHICH IS STANDARD FOR DAVE MITCHELL WP FILES. ALSO NOTE THAT WPRET WORKED OUT SIZE OF FILES WHICH IS SHOWN ALSO.

AFTER FILETYPE CHANGE

THE TWO WP FILES INST-1 & INST-2 NEED NO CHANGING AT ALL TO BE USED. THE REST DO THOUGH AND I USED LESLIE MILBURN'S EXT12.2 TO CHANGE FILETYPES AND ONCE AGAIN PRINTOUT SHOWS RESULT. PLEASE NOTE SYNTAX FOR FILETYPE CHANGE:

```
CHA"PLOTTO-2",T
```

AFTER START ADDRESS CHANGE

AFTER CHANGING FILETYPES I CHANGED START ADDRESSES USING THE FOLLOWING SYNTAX:

```
REL"PLOTTO-2",7AE9
```

THE LAST DIRECTORY PRINTOUT SHOWS RESULT. IF YOU COMPARE IT WITH THE FIRST YOU'LL NOTE IT IS IDENTICAL EXCEPT FOR THE DATA FILE WHICH HAS AN END ADDRESS OF 0100 WITH CORESPONDING FILE SIZE. IT IS NO PROBLEM AS IT LOADED OK WITH NO CORRUPTION EVIDENT IN IT AND ALL OTHER FILES AS WELL.

E & F WP PATCH 3.3: $20.00 PATCH 3.3 WRITTEN BY DAVE MITCHELL WILL CONVERT YOUR E & F TAPE WORD PROCESSOR FOR FULL DISK USE WHILE RETAINING ALL ORIGINAL FUNCTIONS. IT ALSO HAS SHIFT LOCK AND PRINTER CONTROL CODES WHICH CAN BE IMBEDDED IN TEXT AND SAVED TO TAPE OR DISK. **BSTWP.F:** THIS UTILITY PROVIDED WITH PATCH 3.3 WILL CONVERT BASIC PROGRAMS AND ED/ASS. SOURCE CODE FILES INTO WORD PROCESSOR FILES.

DISK FILER CATALOGUER: $25.00 IF YOU HAVE TROUBLE FINDING SOME OF YOUR PROGRAMS THEN THIS NEW DISK CATALOGUEING UTILITY MIGHT BE JUST WHAT THE DISK DOCTOR ORDERED. DISK FILER IS A DATABASE THAT WILL READ, SORT & PRINT YOUR DISK DIRECTORIES. FINDING ANY FILE WILL BE AS EASY AS LOOKING AT A CATALOGUE.

EXTENDED DOS V1.3: $15.00 THESE COMMANDS ARE AT YOUR DISPOSAL: MERGE, DIRA, DIRA, DIRB, LDIRB, OLD, OLD., DEC, HEX, MENU, CODE, LTAB, MOVE AND UPDATE, STATUSA AND LSTATUSA. STATUSA AND LSTATUSA ALSO WORKS WITH VERSION 1.0 DOS

MENU/FILE COPIER — $15.00 THIS UTILITY WILL READ YOUR DISK DIRECTORY AND PRESENT YOU WITH SEVERAL OPTIONS. USING THE CURSOR YOU CAN RUN/BRUN ANY PROGRAM OR SELECT FILE COPY, REN, ERASE, DRIVE 1 OR 2, ETC. BESIDES COPYING TEXT AND BINARY FILES ALL OTHER FILES CAN BE COPIED AS WELL EXEPT FOR DATA FILES.

PRICES INCLUDE POSTAGE - FOR PURCHASE OR INFORMATION CONTACT:
DAVE MITCHELL 24 ELPHINSTONE STREET NORTH ROCKHAMPTON 4701
QUEENSLAND AUSTRALIA - PHONE: (079) 27 8519

CONTRIBUTIONS TO THE JOURNAL

IF YOU ARE THINKING OF CONTRIBUTING TO THE JOURNAL THE PREFERED FORMAT IS BASIC LISTINGS, WORD PROCESSOR OR SOURCE CODE FILES ON TAPE OR DISK. FILES FROM THE FOLLOWING WORD PROCESSORS CAN BE ACCEPTED :-

E & F TAPE OR DISK PATCH 3.1-3.3, WORDPRO CARTRIDGE, WORDPRO PATCH, ALL SOURCE CODE FILES AND ALL QUICKWRITE WORD PROCESSOR FILES.

CLUB MEETINGS — ALL WELCOME

MEETINGS WILL BE APPROXIMATELY ONCE A MONTH. BECAUSE SOME LOCAL MEMBERS WORK SHIFTWORK MEETING DATES WILL BE ADJUSTED TO ACCOMODATE THEM. WHETHER YOU ARE A LOCAL MEMBER, INTRA OR INTERSTATE VISITOR PLEASE CHECK WITH JOE LEON FIRST.

HUNTER VALLEY VZ JOURNAL SUBSCRIPTIONS

SUBSCRIPTION TO - AUST. - 3 ISSUES $11.00 - 6 ISSUES $21.00
H.V.VZ.JOURNAL - N. Z. - 3 ISSUES $13.00 - 6 ISSUES $26.00
PRICES INCLUDE POST & PACKING

FOR MORE INFORMATION ON MEETINGS AND SUBS CONTACT:
JOE LEON 33 TIGHES TCE TIGHES HILL 2297 (049) 692 399 AUSTRALIA

VZ USER GROUPS & PUBLICATIONS

VZ DOWN UNDER & VZ DISK MAGAZINE HAVE CEASED PRODUCTION

WAVZ - WESTERN AUSTRALIA VZ USER GROUP
GRAEME BYWATER PO BOX 388 MORLEY W A 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178
SOFTWARE FOR SALE - DISK MENU

NOTE: WHEN WRITING TO ANY ABOVE OR H.V.VZ. USERS' GROUP FOR INFORMATION PLEASE ENCLOSE A S.S.A.E. OR NZ 2 INT. REPLY COUPONS.

INDEX

JUNE 86 — JUNE 93

HUNTER VALLEY
UZ USERS GROUP
NEWSLETTER

ISSUES 1—4

HUNTER VALLEY
UZ USERS GROUP
MAGAZINE

ISSUES 5—6

HUNTER VALLEY
UZ JOURNAL

ISSUES 7—44

HUNTER VALLEY VZ JOURNAL INDEX 43-01

ISSUE-PAGE	CONTENTS
01-02-02	EDITORIAL
03-03	PRESIDENT'S REPORT
04-04	TECHNICAL REPORT - VZ INPUTS/OUTPUTS BY JOE LEON
05-05	VZ INPUT/OUTPUT CONNECTIONS
06-06	GAMES REVIEW - DAWN PATROL
07-07	UTILITY REVIEW - PROGRAM COPIER
08-09	IS THIS A VIDEO DISK? - BINARY FILE COMMANDS BY ROBERT QUINN
09-09	HI-RES SCREEN DUMP FOR DMP
09-09	SECRETARY'S REPORT
10-11	OH! BOY,NOW IT EVEN TALKS - SPEECH SYNTHESISERS BY DAVE BOYCE
12-12	WORD SEARCH - BY DAVE BOYCE
02-03-03	EDITORIAL
04-04	PRESIDENT'S REPORT
05-0	A CHALLENGE - 2 BASIC GAMES
07-07	GAMES REVIEW - LEARJET
08-08	TECHNICAL REPORT - SUPER II VZ MOD BY JOE LEON
09-10	AND NOW, YOUR OWN ORCHESTRA - COMPUMUSE BY DAVE BOYCE
11-11	HINTS FOR BEGINNERS
12-12	SWAP & SHOP
13-13	COMPUTER CLASSES
03-03-03	PRESIDENT'S REPORT
05-05	PLAY WITH THIS - METEOR BY JAMIE PERRY
06-08	PRINTER BUFFER - BY DAVE BOYCE
09-09	GAMES REVIEW - VZ ASTEROIDS
10-11	MODE(1) PEEK & POKE BY MATTHEW TAYLOR
12-13	TECHNICAL REPORT -18K MEMORY EXPANSION BY JOE LEON
14-16	UNDERSTANDING YOUR VZ - PART 1 BY ROBERT QUINN
17-19	HINTS FOR BEGINNERS
20-20	HINTS FOR DOS USERS
20-20	SWAP & SHOP
20-20	ERRATA
04-03-03	PRESIDENT'S REPORT
04-04	EDITORIAL
05-07	TECHNICAL REPORT - BATTERY BACK UP RAM BY DAVE BOYCE
08-08	SERIAL INTERFACE - DSE TERMINAL
09-09	PRINTER BUFFER 2 BY DAVE BOYCE
10-12	UNDERSTANDING YOUR VZ - PART 2 BY ROBERT QUINN
13-14	FOR BEGINNERS - MATHS TEACHER BY PETER ELLIS
15-15	VZ R.T.T.Y. KIT
16-16	HINTS FOR BEGINNERS
16-16	SWAP & SHOP
05-03-03	PRESIDENT'S REPORT
04-04	EDITORIAL
05-07	TECHNICAL REPORT - VZ200 KEYBOARD MOD BY DAVE BOYCE
08-09	PEEK VS INKEYS
09-10	PRINTER/PLOTTER SECTION BY DAVE BOYCE
11-11	UNDERSTANDING YOUR VZ - PART 3 BY ROBERT QUINN
12-12	GAMES REVIEW - GALAXON
13-13	FOR BEGINNERS - HI-RES DRAWING
14-14	HIGH SCORE TABLE
06-03-03	EDITORIAL
04-04	DISK MENU MAKER #1 BY JOE LEON
05-06	UTILITY REVIEW - HACKER'S DELIGHT
06-06	IMPORT REPORT
07-08	TECHNICAL REPORT - SOFT START INTERRUPT BY DAVE MITCHELL

HUNTER VALLEY VZ JOURNAL INDEX 43-03

ISSUE-PAGE	CONTENTS
13-04-06	HI-RES DRAWING ROUTINES BY DON ISLES
06-06	VZ USER GROUPS
06-08	8K BIB RAM - PART 1 BY JOE LEON
09-10	BASE NUMBER CONVERTER BY DAVE MITCHELL
10-10	MODIFYING VZ200/300 16K RAM PACK BY ROSS WOODS
10-10	SUPER II VZ200 MOD - ETI 687 BY ROSS WOODS
11-13	VZ TOKENS & WORDS - PART 1 BY ROBERT QUINN
14-14	GAMES REVIEW - GALAXON BY PETER J. HILL
15-16	AEM 4505 - TONE CONTROL BY DAVE BOYCE
17-18	DOT MATRIX PRINTERS - PART 1 BY LARRY TAYLOR
19-19	FOR SALE - EXTENDED DOS V1.0
19-19	FOR SALE - VZ DATABASE
20-20	FOR SALE - TAPE TO DISK ED/ASS CONVERTER
20-20	FOR SALE - E&F WP PATCH V3.1
20-20	FOR SALE - QUICKWRITE WP
14-02-02	INDEX
03-03	DISK DRIVE PRIMER
03-03	VZ USER GROUPS
04-04	EDUCATIONAL - ELEMENTS BY PAUL LEON
05-07	CHARACTER CODES BY ROBERT QUINN
07-07	DISK MAILING LIST UPDATE 2
08-10	SPEECH SYNTHESISER PROGRAMS BY DAVE BOYCE
10-10	INKEY$ INPUT ROUTINE BY PAUL LEON
11-13	USING DISK TOKENS - PART 2 BY ROBERT QUINN
13-13	32K BIB RAM BY JOE LEON
14-16	8K BIB RAM - PART 2 BY JOE LEON
17-18	VZ TOKENS & WORDS - PART 2 BY ROBERT QUINN
19-19	FOR SALE - EXTENDED DOS V1.0
19-19	FOR SALE - VZ DATABASE
20-20	FOR SALE - TAPE TO DISK ED/ASS CONVERTER
20-20	FOR SALE - E&F WP PATCH V3.1
20-20	FOR SALE - QUICKWRITE WP
15-02-02	INDEX
03-03	EDITORIAL
03-03	VZ USER GROUPS
04-05	UTILITY REVIEW - PRINTER PATCH V1.4
05-06	EDUCATIONAL - MENTAL MATHS BY JOHN GARLAND
07-07	HI-RES GRAPHICS - KALEIDOSCOPE BY ROBERT QUINN
08-09	HI-RES M/C CSAVE/CLOAD BY DAVE MITCHELL
10-13	DISK MAILING LIST UPDATE 3 BY JOE LEON
14-14	BOOK REVIEW - VZ 200/300 A/L PROGRAMMING MANUAL BY ROSS WOODS
15-16	BLOCK TRANSFERS - PART 1 BY CHRIS HOBROUGH
17-19	ENHANCING FIND - PART 1 BY LARRY TAYLOR
19-19	REPLACING VZ200/300 MEMBRANE KEYBOARD - PART 1 BY JOE LEON
20-20	FOR SALE - EXTENDED DOS V1.0
20-20	FOR SALE - E&F W.P. PATCH V3.1
16-02-02	INDEX
03-03	FOR SALE
03-03	VZ USER GROUPS
04-04	EDUCATIONAL - SPELLING AID BY PAUL & JOE LEON
05-05	IMPROVING VZ SOUND QUALITY BY JOE LEON
06-07	ENHANCING FIND - PART 2 BY LARRY TAYLOR
08-09	BASIC BLOCK TRANSFER - PART 2 BY CHRIS HOBROUGH
10-11	DISK MAILLIST UPDATE 4 BY JOE LEON
12-13	8K BIB RAM - PART 3 BY JOE LEON
14-16	LPRINTER BY ROBERT QUINN
16-16	VZ MEMBRANE KEYBOARD - PART 2 BY JOE LEON

HUNTER VALLEY VZ JOURNAL INDEX 43-05

ISSUE-PAGE	CONTENTS
20-04-05	PAGED DISK MENU BY PAUL & JOE LEON
06-07	SUITE BY ROBERT QUINN
08-08	PRINTER BUFFER UPDATE BY DAVE BOYCE
09-10	32K RAM - 64K EPROM BOARDS BY DAVE NEWCOMBE
11-11	60*60 DIGIT MULTIPLY ROUTINE BY NEVILLE HUGHES
12-13	BEAM HEADING BY DES HOLMES
14-14	4K-64K RAM/EPROM BY JOE LEON
15-15	MUSICAL JOYSTICKS - BY B. GREGG
15-15	FOR SALE - E&F WP PATCH V3.1
16-16	FOR SALE - EXTENDED DOS V1.3
16-16	FOR SALE - DISK MENU/FILE COPIER
16-16	FOR SALE - VZ HI-RES SCREEN EDITOR
21-02-02	INDEX
03-03	VZ NEWS
04-05	ON GOTO BY BRIAN GREEVE
06-06	DISK DRIVE PROBLEMS BY JOE LEON
07-07	DISK DRIVE EXTENSION LEAD BY JOE LEON
08-09	CUSTOMIZING E&F WP BY DAVE BOYCE
10-12	DISK UTILITY BY JOE LEON
13-13	MORTGAGE REPAYMENT BY ROSS WOODS
14-15	128K SIDEWAYS RAM - PART 2 BY J.LEON
16-16	HEAD CLEANING DISK UPDATE BY JOE LEON
17-18	DISK LABELS BY DAVE BOYCE
19-19	VZ USER GROUPS
19-19	FOR SALE - VZ HI-RES SCREEN EDITOR
19-19	HINTS & TIPS - BY JOE LEON
20-20	FOR SALE - E&F WP PATCH V3.3
20-20	FOR SALE - EXTENDED DOS V1.3
20-20	FOR SALE - DISK MENU/FILE COPIER
22-02-02	INDEX
03-03	VZ NEWS
04-05	THE VZ - WHERE TO FROM HERE? BY LARRY TAYLOR
06-07	PRINTER CONTROL CODES BY EDDIE TOMES
08-10	HI-RES DRAWER BY ROBERT QUINN
11-12	DRIVE HEAD PARK ROUTINE BY DAVE MITCHELL
12-12	HINTS & TIPS FOR DISK DRIVE USERS
12-12	HINTS & TIPS FOR TAPE USERS
13-15	HI/LO-RES COPIER BY JOE LEON
16-19	VZ SUPER GRAPHICS BY JOE LEON
19-19	VZ USER GROUPS
20-20	FOR SALE - E&F WP PATCH V3.3
20-20	FOR SALE - EXTENDED DOS V1.3
20-20	FOR SALE - DISK MENU/FILE COPIER
23-02-02	INDEX
03-03	VZ NEWS
04-05	NUMBER SYSTEMS BY LARRY TAYLOR
06-07	TAPE/DISK SOURCE CODE FILE CONVERTER BY BRIAN GREEVE.
08-08	ENHANCING THE LIST COMMAND BY DAVE MITCHELL.
09-11	HI-RES SCREEN MOVE AND PRESERVE UTILITIES BY BOB KITCH.
12-13	HI/LO-RES COPIER PART II BY ROBERT QUINN
14-15	128K SIDEWAYS RAM PART III BY JOE LEON
16-18	VZ SUPER GRAPHICS PART II BY JOE LEON
19-19	VZ USER GROUPS - FOR SALE - SCREEN ED
20-20	FOR SALE - NEW PATCH3.3 - EXT.DOS - MENU/FILE COPIER

CONTINUED NEXT ISSUE